# Teach Them How They Learn: Learning Styles and Information Systems Education

**Casey G. Cegielski**
**Benjamin T. Hazen**
**R. Kelly Rainer**
Department of Management
Auburn University
Auburn, AL 36849, USA
casey@business.auburn.edu

**ABSTRACT**

The rich, interdisciplinary tradition of learning styles is markedly absent in information systems-related research. The current study applies the framework of learning styles to a common educational component of many of today's information systems curricula - object-oriented systems development - in an effort to answer the question as to whether one's learning style, when matched with a specific complementary instructional methodology, results in increased domain-specific performance. The data collected from 196 information systems majors enrolled in object-oriented systems development courses suggest that task performances increases significantly when the instructional methodology closely mirrors the student's learning style inclination.

**Keywords**: Learning Styles, Object-Oriented Programming, Unified Modeling Language (UML), Computer Programming

## 1. INTRODUCTION

There are many differing contexts associated with the word "style." In a general context, style may refer to the characteristics which signify, unify, or distinguish an entity via form or function (Merriam-Webster, 2008). It is common to describe and classify unique styles in many domains. For example, there are various architectural styles that may be classified by elements of form, material, time period, and indigenous geographic region. Similarly, there are many distinct literary styles, classified by form, genre, and technique. However, style is not a term that is particularly well-associated with the processes that comprise the complex mechanism of individual learning. However, recent research suggests that the style by which one learns and applies knowledge is an important characteristic to consider in the aggregate educational processes (Graf, Lin and Kinshuk, 2008; Kolb and Kolb, 2009; Syler et al., 2006; Thorton, Haskell and Libby, 2006; Zualkernan, Allert, and Qadah, 2006)

Acknowledgement of unique learning styles is an attempt to characterize the complex processes by which one acquires knowledge (Kolb, Rubin and McIntyre, 1974). Learning style may be thought of as a formulation of preconceptions by an individual engaged in the activity of learning (Biggs and Moore, 1993). These preconceptions may include a combination of one's expectations based on previous experiences, one's cognitive ability, and one's personality (Hall, Cegielski and Wade, 2006; Kiguwa and Silva, 2007). The literature in the area of learning styles indicates that some individuals demonstrate a more rapid absorption of subject matter when the pedagogical approach utilized in instruction closely mirrors the students learning style inclination (Felder and Silverman, 1988; Garcia, Schiaffino and Amandi, 2008; Honey and Mumford, 1992; Kolb, 1984; Litzinger and Osif, 1993; Park et al., 2010).

The motivation for the current study is very specific with regard to the aforementioned assertion. The current study assesses the instructional impact of a treatment designed to facilitate the learning of object-oriented systems development (OOSD) for students who previously demonstrated an inclination towards a visual learning style. From the results of this study, instructors engaged in the teaching of OOSD may better utilize knowledge regarding learning styles as a tool to enhance student performance.

The remainder of this manuscript is arranged in the following manner. First, we describe the significant concepts associated with learning styles; an area of popular pedagogical research that is heretofore underrepresented in information systems (IS) development research. Next, we describe the unique aspects of teaching OOSD. Thus, we present a comprehensive review of the general literature of learning styles and subsequently discuss the applications of such concepts to IS development through the contextual perspective of object-oriented programming languages (OOPL). Then, we propose a model that serves as the framework for this study and the basis of hypothesis development. Next, we discuss the methods employed to 1) assess the learning styles of the study participants and 2) measure the affect on outcomes via the treatment applied to the subjects. Finally, we present our findings as they relate to education of students engaged in OOSD courses.

## 2. LITERATURE REVIEW

### 2.1 Learning Styles

Learning is a predominant cognitive function in human beings, which drives the development of new capacities, skills, values, understanding, and preferences (Yannakakis, Maragoudakis and Hallam, 2009). We define learning as the acquisition of different types of knowledge through the assimilation of data via the five senses. Although the definition is concise, the construct of learning is multi-faceted (Saljo, 2009). Review of the published literature on learning reveals several substantial areas of active investigation related to the activity. For example, many researchers have explored the factors that affect one's capacity to learn (Claxton, 2007). Additionally, recent research has focused on one's desire or motivation to learn (Dreher et al., 2009; Shroff, Vogel and Coombes, 2008; Yair, 2000; Wang and Braman, 2009). Still, others investigate various aspects of how people learn (Klasnja-Milicevic et al., 2011; Strickland et al., 1991; Wang and Liao, 2011). It is this last area of research in which the aspects of learning styles are grounded.

Although there is significant depth of research and interest in learning styles in both the educational and the psychological domains, applied empirical investigation of learning styles within the IS domain is lacking. For IS researchers and educators, it should be obvious that there is value in addressing the void that exists in both research and pedagogical areas of IS as it relates to learning styles. Through this study, we seek to address this gap by providing an initial foundation from which other researchers may contribute to the development of a rich research tradition of learning styles in the IS domain.

A *learning style* is an aggregate construct of cognitive, affective, and psychological factors that provide insight into how an individual responds to a specific pedagogy (Kolb, 1984). Research in learning styles theory suggests that each individual has an inclination towards a particular multi-faceted modality for learning (Cagiltay, 2008). The theory identifies four different constructs as the foundation of one's learning style, as summarized in Table 1. Individuals are categorized by their level of engagement during the learning process (active-reflective), affinity for abstraction (sensing-intuitive), preferred input methodology (visual-verbal), and perceptual capabilities (sequential-global) (Felder and Silverman, 1988). Every individual has some inclination in each of the aforementioned four dimensions. For example, one may favor learning through group work using pictures and diagrams to organize concrete facts into a series. This person would be classified, according to the theory, as an *active-sensing-visual-sequential* learner.

The literature in the research domain of learning styles suggests that the process of learning is facilitated more aptly when the instructional methods match the learner's style inclination (Allinson and Hayes, 1996; Felder and Brent, 2005; Hsieh et al., 2011). Simply stated, one may acquire a better understanding of the subject matter in question when one engages in a learning activity that functionally mirrors one's own dominant learning style. From a practical educational perspective, it is difficult for an educator to possess a priori knowledge of his or her student's learning styles. However, this phenomenon belies our research question: does teaching to one's learning style improve one's performance? This is the overarching question that, when put into the context of IS, serves as the motivation for our study.

There are a number of different learning style assessment tools (Graf, Lin and Kinshuk, 2008). A frequently utilized instrument, and one designed for use with engineering students in higher education, is the Soloman-Felder Index of

| Construct | Description | Example |
|---|---|---|
| Active-Reflective | The manner in which one engages in processing information | Active learners prefer to engage in group discussions and apply information to common situations<br><br>Reflective learners prefer to cogitate and internally process new information |
| Sensing-Intuitive | The extent to which one is inclined to embrace concrete or abstract forms of information to form a frame of reference for learning | Sensing learners prefer the empirical facts and tangible work<br><br>Intuitive learners prefer theories and rely on their ability to identify general relationships |
| Visual-Verbal | The degree to which one favors either visual or textual input as the primary input mode in the learning process | Visual learners prefer to use pictures, diagrams, and charts in the learning process<br><br>Verbal learners prefer textual input (written or spoken) of information in the learning process |
| Sequential-Global | The degree to which one prefers the presentation of information in an incremental linear series or a holistic broad strokes | Sequential learners are inclined to apply a stepwise approach to assimilating new information perhaps recognizing the "big picture" after comprehending the underlying components of the information<br><br>Global learners more readily grasp the "big picture" but often miss the details that support the overall message of the information |

**Table 1: Description of the Constructs Associated with Learning Styles**

Learning Styles (ILS) (Felder and Spurlin, 2005; Garcia, Schiaffino and Amandi, 2008). The instrument is comprised of 44 multiple choice questions; 11 questions for each of the four previously discussed learning styles domains. The validity and reliability of the ILS has been established across multiple domains (Felder and Spurlin, 2005; Litzinger et al., 2007; Zywno, 2003). Specifically, and of direct relevance to the current study, the instrument has been utilized several times in technology-based studies (Park et al., 2010; Zualkernan, Albert and Qadah, 2006). Therefore, we deemed ILS to be an appropriate tool to facilitate data collection for our study.

**2.2 Object Oriented Systems Development**

OOSD is a skill valued by employers of IS professionals and, as a result, is a cornerstone course offering of many university- and college-level technology-based business degrees (Hall, Cegielski and Wade, 2006). However, there is very little current academic research on the subject. In fact, it was during the 1970s and 1980s that the majority of research oriented toward understanding the scope and nature of the relationships within the human-computer interface of systems development appeared in academic publications (Cegielski and Hall, 2006.).

Among previously published academic reports, there are several studies in which researchers assessed predictive relationships between an individual's personal attributes (i.e. personality, cognitive ability) and his or her capacity to successfully complete systems development tasks. However, none of the published research reviewed directly examined the affect of learning styles and instructional methods on systems development task performance. Furthermore, most of the published research did not include object-oriented computer languages as a systems development tool.

In an effort to provide a generally broad perspective in relation to IS curricula in higher education, we chose to use OOPL as the contextual artifact in this study. OOPL was chosen as an operationalization of systems development because the languages are pervasive in IS programs of higher education and present unique educational challenges to both instructors and students alike (Ramesh and Wu, 2004). For the educator, it is often difficult to effectively teach the abstract concepts of encapsulation, inheritance, and polymorphism (the foundation of OOPL) while simultaneously instructing students in the syntactic nuances of OOPL (Manns and Nelson, 1996).

A cursory review of currently available textbooks in the OOPL domain reveals that the content offerings of OOPL textbooks often follow either a *sequential* or *parallel* approach to instruction. That is, the textbooks and subsequently designed pedagogy present the concepts of OOPL and the syntactic illustrations of code in a disconnected sequential fashion. For example, the first several chapters of the text are dedicated to concepts and the later chapters of the text address syntax and code development. Conversely, a number of textbooks attempt to intersperse concepts and syntax equally throughout each chapter. In some respects, this method provides a more integrated approach to the instruction and learning of OOPL. However, this approach requires both the instructor and the student to posses the ability to toggle between streams of abstraction and tangible application (Ramesh and Wu, 2004). This can be a challenge for students, particularly because it requires one to apply the concepts independently from one another. In fact, the basic OOPL concepts of encapsulation, inheritance, and polymorphism (which are defined in Table 2) are interdependent to such a degree that one of the concepts cannot be appropriately applied without utilizing, at least tangentially, the two other related concepts. Thus, it would seem that both of these methods of instruction have merit as well as shortcomings.

Although challenges exist for both educators and students engaged in the instruction and the learning of OOPL, it is not the intention nor the focus of the research presented herein to engage a debate on the topic of structure regarding the teaching of OOPL. The aforementioned points are made solely to support the assertion that the instruction of students in OOPL is a very complex process because of the nature of the material, which is highly conceptual as well as practically applicable. Given the complexities of teaching and learning OOPL, we assert that it is important to explore the entire spectrum of factors that may affect a student's ability to digest the knowledge required to apply OOPL skills in a meaningful problem-solving manner.

In industry, object-oriented programming and using tools such as C++ or Java derive much of their popular appeal from the set of unique conceptual attributes that grounds the practical development processes in these types of languages (Fedorowicz and Villeneuve, 1999; Hall, Cegielski and Wade, 2006). Flexible and reusable code is an end result of the application of the concepts that underpin OOPL (Coad and Yourdon, 1991a, 1991b). Specifically, the differences between an OOPL and traditional procedural programming language may be characterized and measured with two metrics – degree of *cohesion* and degree of *coupling* (Booch, 1991; Bradley, 1992; Fenton and Pfleeger, 1997).

In traditional procedural programming languages, a computer program is a set of *interdependent* (coupled) procedures operating on data in the services of a particular goal. Conversely, in OOPL, a program is a set of *independent* autonomous objects that exchange data to fulfill a unified (cohesion) purpose (Cho and Kim, 2002). Functionally, OOPL evolved from the established logic used in procedural languages – top-down modularity (Pennington,

| **Characteristic** | **Definition** | **Operational Expression** |
|---|---|---|
| Encapsulation | The packaging of programming code into wholly independent, self-contained units | Creation of classes |
| Inheritance | The use of existing coded classes as foundational components for the creation of new programming code | The parent-child relationship of extend classes |
| Polymorphism | The capability of an object to retain a generalized purpose while assuming different forms of application in separate instances of programming code | Instantiation of autonomous objects from classes |

**Table 2: Object-Oriented Systems Development Concepts(derived from Hall, Cegielski, and Wade, 2006)**

Lee and Rehder, 1995). A greater degree of modularity through less *coupling*, interdependency between program routines, and greater *cohesion* facilities the development of highly *flexible* and *reusable* software (Fichman and Kemerer, 1992; Hall, Cegielski and Wade, 2006). As a point of reference, highly coupled program routines are indivisible because each routine is dependent upon the function of another routine (White, 2003). A computer program that exhibits a high degree of coupling is more difficult to maintain, to extend, and to reuse due to extensive internal interdependences among routines (Kolling, 1999). However, a computer program that exhibits a high degree of cohesion consists of elements that are separable, and thus independent in scope, from an aggregate program of which they were only a component. In this form, components are reusable and, therefore, flexible (Sultan and Chan, 2000). For clarity, Figure 1 illustrates the operationalization of systems development via OOPL.

## 3. RESEARCH MODEL AND HYPOTHESES

As noted in the previous summary of learning styles literature, there is some indication that the coordination of a student's type of learning style with an instructional method that closely reflects that learning style may enhance measurable outcomes in the educational process (Allinson and Hayes, 1996; Felder and Brent, 2005; Lau and Yuen, 2011; Prajapati et al., 2011). From this assertion, we constructed a general conceptual model, which is depicted in Figure 2. In the model, instructional method mediates the relationship between individual learning style and educational outcomes. Our study operationalizes this model via investigating a specific IS instructional tool, Unified Modeling Language (UML), and programming performance outcomes. Thus, Figure 2 doubles as our research model. As the model suggests, our study investigates whether or not the coordination between one's learning style and a similar instructional methodology has an impact on performance outcomes.

In order to facilitate testing of the model, it was necessary to develop appropriately broad task performance

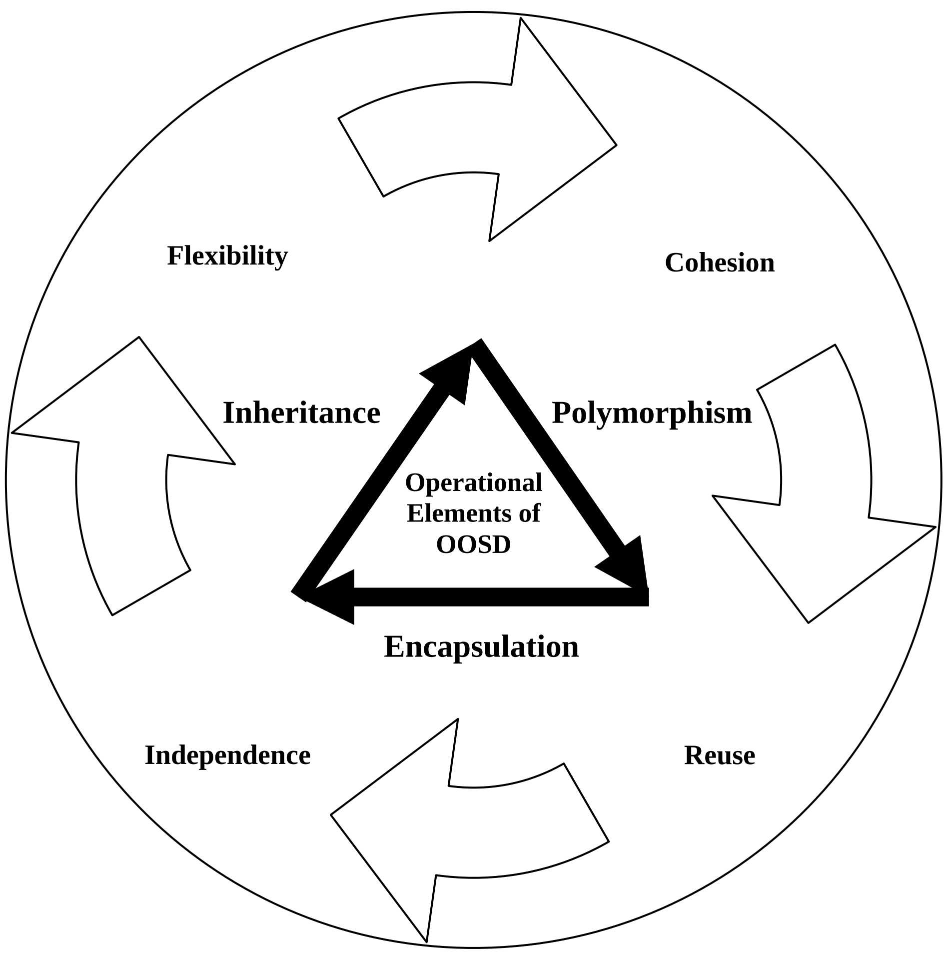



**Figure 1: Implementing Operational Elements of Object Oriented Systems Development**

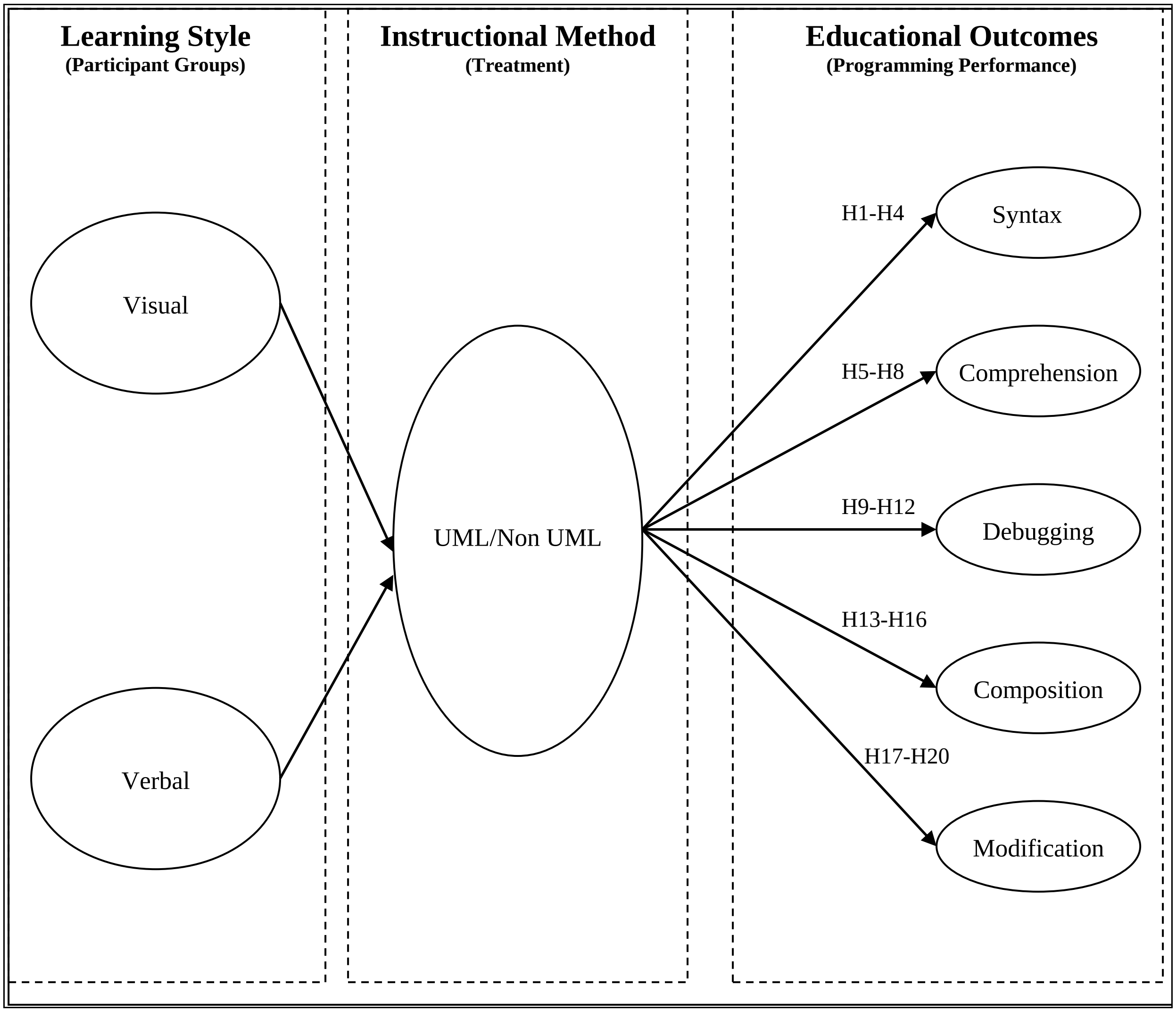



**Figure 2: Research Model in Reference to Conceptual Framework**

measures for OOSD. A review of the literature produced 24 articles published between 1971 and 2008 that reported dependent variables related to computer programming skills and ability. Overwhelmingly, the dependent variables in most of the aforementioned studies were in the form of some sort of grade. While common in the literature, course grade or a single exam score are not appropriately granular dependent variables to support the analysis of the research question posited herein. Therefore, we opted to operationalize the measurement of object-oriented programming performance through a series of tests that assess one's ability in the domain from a multidimensional perspective.

Foreman (1988) argued that computer programming performance was most aptly assessed through a distillation of the interrelated components that comprise the entirety of the activity. According to Foreman (1988), the components that comprise the activity of computer programming are 1) *syntax*, 2) *comprehension*, 3) *debugging*, 4) *composition*, and 5) *modification.* Thus, we adopt these specific components as the dependent variables employed in our current study. Operational definitions and contextual examples of these variables are illustrated in Table 3.

The research model suggests 20 testable relationships, which invoke the following hypotheses.

**H1**: Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *syntax test* than those students who demonstrate an inclination toward a *visual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H2**: Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *syntax test* than those students who demonstrate an inclination toward a v*erbal learning style* and are not

| Component | Definition | Example |
|---|---|---|
| Syntax | A demonstration of the mastery of the grammatical rules of the language | The appropriate use of assignment operators in the declaration of objects |
| Comprehension | One's ability to read and understand the functional aspects of a program | The ability to predict programming outputs from given inputs |
| Debugging | The ability to identify errors found within existing code and offer potential solutions | The ability to recognize that variables shared among methods must be global |
| Composition | One's capability to write functionally complete code | Generating a functional program from scratch |
| Modification | The capability to edit existing code so as to change the function of said code | Rewriting a method so that the new method will allow the use of user input as opposed to system data |

**Table 3: Summary of the Component Activities of Computer Programming**

instructed using a tool that supports a verbal learning style instructional method.

**H3**: Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *syntax test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H4**: Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *syntax test* than those students who demonstrate an inclination toward a v*isual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H5:** Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *debugging test* than those students who demonstrate an inclination toward a *visual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H6:** Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *debugging test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H7**: Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *debugging test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H8**: Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *debugging test* than those students who demonstrate an inclination toward a v*isual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H9:** Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *composition test* than those students who demonstrate an inclination toward a *visual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H10:** Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *composition test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H11**: Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *composition* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H12**: Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *composition test* than those students who demonstrate an inclination toward a v*isual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H13:** Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *comprehension test* than those students who demonstrate an inclination toward a *visual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H14:** Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *comprehension test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H15**: Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *comprehension test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H16**: Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *comprehension test* than those students who demonstrate an inclination toward a v*isual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H17:** Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *modification test* than those students who demonstrate an inclination toward a *visual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

**H18:** Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *modification test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H19**: Students who demonstrate an inclination toward a *visual learning style* and are instructed using a tool that supports a visual learning style instructional method, will perform better on an object oriented computer programming *modification test* than those students who demonstrate an inclination toward a *verbal learning style* and are not instructed using a tool that supports a verbal learning style instructional method.

**H20**: Students who demonstrate an inclination toward a *verbal learning style* and are instructed using a tool that supports a verbal learning style instructional method, will perform better on an object oriented computer programming *modification test* than those students who demonstrate an inclination toward a v*isual learning style* and are not instructed using a tool that supports a visual learning style instructional method.

## 4. METHODOLOGY

Over a four year span, data were collected from 196 IS majors who completed an OOSD course offered in the college of business at a national university in the southeastern United States. The 16 week course is required for IS majors and open to those students with junior or senior standing and a university-validated overall grade point average of 2.20 or higher. During the data collection period, multiple sections of the course were offered. Over the term of the data collection period, the scope of the content presented was identical and each section was led by the same instructor. Each course met twice a week for one hour and fifteen minutes. There were no additional meetings of the courses beyond the 40 hours of instructional contact required by the institution for a three credit hour course.

An analysis of the demographic information collected on the subjects that participated in the current study revealed a relatively homogenous sample. Seventy-one percent of the subjects were males and 92% of those who participated in the study were between the ages of 18 and 25 years of age. On the first day of class during each semester, each of the students was asked to complete the 44 question Soloman-Felder ILS. From an analysis of this data, students were initially classified as either visual or verbal learners. The ILS administration was repeated at the end of the semester as a means to ensure a reliable classification of the students. There were no reclassifications of students based on the replication of the ILS and the subsequent analysis of the data.

### 4.1 Experimental Treatment: Unified Modeling Language

UML, introduced in 1997 and widely used in software development, is a standardized, general purpose modeling convention that consists of 13 diagram types that facilitate the visual presentation of abstract object oriented computer programming concepts (Dzidek, Arisholm and Briand, 2008). In practice, UML diagrams are often the first artifacts of a computer software architecture from which a system is subsequently created (Lange, Chaudron, and Muskens, 2006). UML was selected as a treatment for this experiment because it presents a *visual* illustration through which object-oriented computer code may be represented. Although there are other modeling tools that may have served aptly as a treatment in this experiment, UML offers several particular advantages over the other possible alternatives.

First, because only 13 general artifacts are used, UML is generally easy for students to learn and employ. In this experiment, a subset of only three of the artifacts were used: class level, method level, and object level. Second, the tool is designed specifically for use in modeling object oriented systems. This was a particularly important factor as the systems development process in question is object oriented. Finally, UML, unlike many other modeling tools, requires no investment of additional software to facilitate its use. This eliminates a potential resource constraint for both the instructor and the students.

As reported in Table 4, UML was the single treatment applied to 95 of the 196 subjects; the remaining 101 students received no UML. During the first week of each semester, the 95 UML subjects were presented with an overview of UML and shown how to read and diagram object oriented computer code using the aforementioned tool. Additionally, these 95 students were provided a UML diagram for each of the 16 teaching examples utilized in the course. Before discussing a teaching example during a lecture, the instructor visually presented, via an electronic white board, the UML diagram and related the example to the entire class.

| Student Learning Style | UML | Non UML | Total |
|---|---|---|---|
| **Visual** | 56 | 63 | 119 |
| **Verbal** | 39 | 38 | 77 |
| **Total** | 95 | 101 | 196 |

**Table 4: Distribution of Subjects in Treatment Groups**

## 5. RESULTS

Each of the 20 research hypotheses was assessed using ANOVA. The summary of the findings for each statistical test are presented in Table 5. Hypothesis **H1**, **H5**, **H9**, **H13**, and **H17** were constructed to assess whether or not visual learners perform better when instructed with a pedagogy that is tailored toward a visual learning style. The findings support each of the aforementioned hypotheses that visual learners will perform statistically better on OOSD tasks than those visual learners who are not instructed using a pedagogy that supports a visual learning style.

Results of hypothesis **H10**, **H14**, and **H18**, which were constructed to assess whether or not verbal learners perform better when instructed with a pedagogy that is tailored toward a verbal learning style, provide some support for the assertion. Specifically, verbal learners instructed through a verbal learning style demonstrated no statistically significant difference in performance from other verbal learners who were instructed with the visual learning style treatment. This provides support for the assertion that visual learning style instruction may not produce measurable benefits when utilized with verbal learners.

Two hypotheses that were also constructed to assess whether or not verbal learners perform better when instructed with a pedagogy that is tailored toward a verbal learning style, **H2** and **H6,** were not supported. Neither were **H3** and **H7** supported. However, when comparing the performance of visual and verbal learners who both received instruction that incorporated the visual learning treatment of UML, there is a significant statistical difference among groups regarding the dependent variables of c*omprehension*, *composition*, and *modification* (**H11**, **H15**, and **H19**). However, it is possible that the dependent variables in question for each of the non-supported hypotheses (*syntax* and *debugging*) do not exhibit as strong a correlation to learning style as *comprehension*, *composition*, and *modification*.

Finally, there was no statistically significant difference for any of the dependent variables assessed through a comparison of visual and verbal learners where neither group received the visual treatment (**H4**, **H8**, **H12**, **H16**, **and H20**). These hypotheses represented the control groups in the study. Because of the lack of significant statistical difference among these groups, our assumption that the instruction method and presentation of the material was consistent across sections is supported.

## 6. DISCUSSION AND IMPLICATIONS

Our research investigates whether one acquires a better understanding of the OOSD knowledge when one engages in a learning activity that functionally mirrors one's own dominate learning style. We now discuss the findings and implications of our investigation into this topic.

### 6.1 Visual-Oriented Instruction Enhances Performance of Visual Learners

The current experiment provides support for the general conclusion that a student who demonstrates an inclination toward a visual learning style will perform better on subject specific tasks if he or she is instructed on those tasks using a visual teaching tool. In our study, we found a clear difference between the two groups utilized to assess this assertion. In all five outcome measures, *syntax, debugging*, *comprehension*, *composition,* and *modification*, students classified as visual learners when instructed with the visual treatment tool of UML performed significantly better than those visual learns who were instructed without the visual treatment of UML. Additionally, the data provide evidence for the assertion that students with a visual learning style perform better on specific object-oriented programming tasks than students with a verbal learning style when both groups are instructed in visual learning style.

Regarding visual learners and visual instructional style, the results of three of the five specific measurement outcomes, *comprehension*, *composition*, and *modification,* support the conclusion that visual learners that are instructed with a visual learning style perform better on tasks specific to the knowledge domain than verbal learners who also

| Hypothesis | Outcome Measure | Learning Style Comparison | Treatment | P-Value | Conclusion |
|---|---|---|---|---|---|
| H1: | Syntax | Visual/Visual | UML/No UML | 0.002 | Supported |
| H2: | Syntax | Verbal/Verbal | No UML/UML | 0.474 | Not Supported |
| H3: | Syntax | Visual/Verbal | UML/UML | 0.081 | Not Supported |
| H4: | Syntax | Verbal/Visual | No UML/No UML | 0.051 | Not Supported |
| H5: | Debugging | Visual/Visual | UML/No UML | <0.001 | Supported |
| H6: | Debugging | Verbal/Verbal | No UML/UML | 0.385 | Not Supported |
| H7: | Debugging | Visual/Verbal | UML/UML | 0.441 | Not Supported |
| H8: | Debugging | Verbal/Visual | No UML/No UML | 0.189 | Not Supported |
| H9: | Composition | Visual/Visual | UML/No UML | <0.001 | Supported |
| H10: | Composition | Verbal/Verbal | No UML/UML | 0.413 | Not Supported |
| H11: | Composition | Visual/Verbal | UML/UML | 0.009 | Supported |
| H12: | Composition | Verbal/Visual | No UML/No UML | 0.544 | Not Supported |
| H13: | Comprehension | Visual/Visual | UML/No UML | <0.001 | Supported |
| H14: | Comprehension | Verbal/Verbal | No UML/UML | <0.001 | Supported |
| H15: | Comprehension | Visual/Verbal | UML/UML | 0.026 | Supported |
| H16: | Comprehension | Verbal/Visual | No UML/No UML | 0.051 | Not Supported |
| H17: | Modification | Visual/Visual | UML/No UML | <0.001 | Supported |
| H18: | Modification | Verbal/Verbal | No UML/UML | 0.029 | Supported |
| H19: | Modification | Visual/Verbal | UML/UML | 0.001 | Supported |
| H20: | Modification | Verbal/Visual | No UML/No UML | 0.048 | Not Supported |

**Table 5: Summary of Hypothesis Test Results**

receive the visual UML treatment. These findings suggest that visual learners may derive an additional benefit from instruction with visual instruction of UML, whereas verbal learners may not derive a benefit from the UML treatment.

In summary, we conclude: 1) among visual learners, those subjects instructed with a visual tool perform better than those subjects who were not instructed with a visual tool and, 2) when both visual and verbal learners are instructed with the same visual tool, the visual learners perform better than the verbal learners. These findings are important in that they provide empirical support within the knowledge domain of IS for what has been expressed in the general educational and psychological literature regarding learning styles and performance.

### 6.2 Verbal-Oriented Instruction Enhances Performance of Verbal Learners

The findings in the current experiment offer support for the assertion that students with a verbal learning style inclination perform better on specific object-oriented programming tasks when they are instructed in a manner that mirrors a verbal learning style. Specifically, the findings provide support for this assertion in three of the five specific measurement outcomes – *comprehension*, *composition*, and *modification*. Additionally, there were no differences detected in the performance among the groups of visual and verbal learners when no treatment was applied to either group. These findings complement the conclusions developed in the previous discussion about visual learners. This also suggests that verbal learners may not derive any additional benefit from visual instruction.

## 7. LIMITATIONS AND FUTURE RESEARCH

Although the experiment was conducted in a rigorous methodological fashion, nevertheless, there are limitations worth noting. However, it is within the limitations of this study that other researchers with interests in learning styles may begin to develop a broad research agenda to address the gap that exists in the current IS literature.

Similar to the intricacies of human personality, learning styles are complex and often present overlapping dimensions when assessed via a standardized instrument such as the ILS. In this study, students were classified as visual or verbal learners using the 44 question ILS. Most students, based on the individual analysis of the ILS scores, did exhibit a strong inclination toward a given learning style. However, there were six students in the subject pool who did not exhibit a strong inclination toward either the visual or the verbal style. In these cases, the students were classified into the subject category towards which they demonstrated the highest affinity. For example, four students in the subject pool scored high on both visual and verbal sections of the ILS. These students were classified into the visual subject group based on the fact that they exhibited a stronger association towards a visual learning style than verbal learning style. These cases represented 3% of the total subject pool of 196.

In this study, we focus on the *visual-verbal* dimension of learning style. Future research should include an analysis of all four learning styles within the subject domain of IS development. If the research in the area of learning styles within IS is to develop and mature to a point where there exists a complete taxonomy, then the additional dimensions of learning styles need to be addressed.

Although the five dependent variables assessed in the current study present a specific and encompassing perspective of task performance in OOSD, the metrics employed represent only one possible set of assessment criteria. Certainly, it would be necessary in future research to expound upon the metrics used in the current study to provide a greater opportunity to argue that any findings may be generalizable beyond each specific study.

## 8. CONCLUSION

The results of this research provide an initial empirical assessment of learning styles and their relationship to performance in an OOSD course. The contribution of our research is twofold. First, the research presented herein may serve as foundation from which to launch a detailed research agenda in the area of learning styles within the IS educational domain. As a research and educational domain, IS is interdisciplinary in nature. Underlying concepts are derived from multiple domains, to include cognitive psychology, communications, and educational pedagogy. Thus, IS researchers engaged in teaching activities should feel comfortable delving into the reference disciplines upon which learning styles are grounded. It is a practice that is particularly important if we have the desire to expand the boundaries of the IS domain. Our study serves as an initial step toward expansion in this area.

Most importantly, the findings provide support for the assertion that classroom instruction, when tailored toward a student's learning style, may produce better task performance. As IS curriculum continues to evolve with the ever-changing needs of the global workforce (Apigian and Gambill, 2010; Stefanidis and Fitzgerald, 2010), we undoubtedly want to do our best to effectively deliver relevant content to ensure our students perform well both in the classroom and beyond. As demonstrated in this experiment, one could plausibly foresee an educator with an understanding of learning styles utilize specific classroom techniques that emphasize a particular learning style given his or her student's a priori demonstration of an inclination toward a given learning style. This would certainly require a degree of initiative and preemptive participation on the part of an instructor as well as his or her students. However, should we as educators accept the challenge to ***teach them how they learn***, our impact as educators may be greater and our students may be better prepared for the rigors of the profession into which they aspire to enter.

## 9. REFERENCES


Allinson C. and Hayes, J. (1996), "The Cognitive Style Index: A Measure of Intuition Analysis for Organizational Research." Journal of Management Studies, Vol. 33, No. 1, pp. 119-135.

Apigian, C. H. and Gambill, S. E. (2010), "Are We Teaching the IS 2009* Model Curriculum?" Journal of Information Systems Education, Vol. 21, No. 4, pp. 411-420.

Biggs, J. and Moore, P. (1993), The Process of Learning. Third edition. Prentice Hall, New York.

Booch, G. (1991), Object-Oriented Design with Applications. Benjamin/Cummings Publishing Company, Redwood City, CA.

Bradley, S. (1992), Object-Oriented Technology and Software Change Management in Distributed Environments. Addison-Wesley, Boston, MA.

Cagiltay, N. E. (2008), "Using Learning Styles Theory in Engineering Education." European Journal of Engineering Education, Vol. 33, No. 4, pp. 415-424.

Cegielski, C. and Hall, D. (2006), "What Makes a Good Programmer?" Communications of the ACM, Vol. 49, No. 10, pp. 73-75.

Cho, I. and Kim, Y. (2002), "Critical Factors for the Assimilation of Object-Oriented Programming Languages." Journal of Management Information Systems Vol. 18, No. 3, pp. 125-156.

Claxton, G. (2007), "Expanding Young People's Capacity to Learn." British Journal of Educational Studies, Vol. 55, No. 2, pp. 115-134.

Coad, P. and Yourdon, E. (1991a), Object-Oriented Analysis, Prentice Hall, Englewood Cliffs, NJ.

Coad, P. and Yourdon, E. (1991b), Object-Oriented Design, Prentice Hall, Englewood Cliffs, NJ.

Dreher, C., Reiners, T., Dreher, N., and Dreher, H. (2009), "Virtual Worlds as a Context Suited for Information Systems Education: Discussion of Pedagogical Experience and Curriculum Design with References to Second Life." Journal of Information Systems Education, Vol. 20, No. 2, pp.211-224.

Dzidek, W. J., Arisholm, E., and Briand, L. C. (2008), "A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance." IEEE Transactions on Software Engineering, Vol. 34, No. 3, pp. 407-432.

Fedorowicz, J. and Villeneuve, A. (1999), "Surveying Object Technology and Benefits: A Test of Conventional Wisdom." Information & Management, Vol. 35, No. 6, pp. 331-344.

Felder, R. and Brent, R. (2005), "Understanding Student Differences." Journal of Engineering Education, Vol. 94, No. 1, pp. 57-72.

Felder, R. and Silverman, L. (1988), "Learning and Teaching Styles." Journal of Engineering Education, Vol. 78, No. 7, pp. 674-681.

Felder, R. and Spurlin, J. (2005), "Application, Reliability, and Validity of the Index of Learning Styles." International Journal of Engineering Education, Vol. 21, No. 1, pp. 103-112.

Fenton, N. and Pfleeger, S. (1997), Software Metrics: A Rigorous and Practical Approach, PWS Publishing, Boston, MA.

Fichman, R. and Kemerer, C. (1992), "Object-Oriented and Conventional Analysis and Design Methodologies." IEEE Computer, Vol. 25, No. 10, pp. 22-29.

Foreman, K. (1988), "Cognitive Style, Cognitive Ability, and the Acquisition of Initial Computer Programming Competence." Proceedings of the Annual Meeting of the Association For Education Communications and Technology, New Orleans, LA.

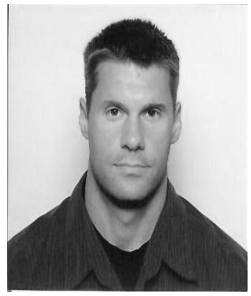
Garcia, P., Schiaffino, S. and Amandi, A. (2008), "An Enhanced Bayesian Model to Detect Students' Learning

Styles in Web-based Courses." Journal of Computer Assisted Learning, Vol. 24, No. 4, pp. 305-315.
Graf, S., Lin, T., and Kinshuk (2008), "The Relationship between Learning Styles and Cognitive Traits - Getting Additional Information for Improving Student Modeling." Computers in Human Behavior, Vol. 24, No. 2, pp. 122-137.
Hall, D. Cegielski, C., and Wade, J. N. (2006), "Theoretical Value Belief, Cognitive Ability and Personality as Predictors of Student Performance in Object-Oriented Programming Environments." Decision Sciences Journal of Innovative Education, Vol. 4, No. 2, pp. 237-257.
Honey, P. and Mumford, A. (1992), The Manual of Learning Styles, Third Edition, Peter Honey Publications, Maidenhead, UK.
Hsieh, S-W., Jang, Y-R., Hwang, G-J., and Chen, N-S. (2011), "Effects of Teaching and Learning Styles on Students' Reflection Levels for Ubiquitous Learning." Computers & Education, Vol. 57, No. 1, pp. 1194-1201.
Kiguwa, P. and Silva, A. (2007), "Teaching and Learning: Addressing the Gap through Learning Styles." South African Journal of Psychology, Vol. 37, No. 2, pp. 354-360.
Klasnja-Milicevic, A., Vesin, B., Ivanovic, M., and Budimac, Z. (2011), "E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification." Computers & Education, Vol. 56, No. 3, pp. 885-899.
Kolb, A. Y. and Kolb, D. A. (2009), "The Learning Way: Meta-Cognitive Aspects of Experiential Learning." Simulation & Gaming, Vol. 40, No. 3, pp. 297-327.
Kolb, D., Rubin, I., and McIntyre, J. (1974), Organizational Psychology: A Book of Readings. Prentice Hall, Englewood Cliffs, NJ.
Kolb, D. (1984), Experimental Learning: Experience as a Source of Learning and Development, Prentice Hall, New York..
Kolling, M. (1999), "The Problem of Teaching Object-Oriented Programming." Journal of Object-Oriented Programming, Vol. 11, No. 8, pp. 8-15.
Lange, C., Chaudron, M., and Muskens, J. (2006), "In Practice: UML Software Architecture and Design Description." IEEE Software, Vol. 23, No. 2, pp. 40-46.
Lau, W. and Yuen, A. (2011), "Modelling Programming Performance: Beyond the Influence of Learner Characteristics." Computers & Education, Vol. 57, No. 1, pp. 1202-1213.
Litzinger, M. and Osif, B. (1993), Accommodating Diverse Learning Styles: Designing Instruction for Electronic Information Sources, In What is Good Instruction Now? Library Instruction for the 90s. Pierian Press, Ann Arbor, MI.
Litzinger, T., Lee, S., Wise, J., and Felder, R. (2007), "A psychometric study of the index of learning styles." Journal of Engineering Education, Vol. 96, No. 4, pp. 309-319.
Manns, M. L. and Nelson, H. J. (1996), "Retraining Procedure-Oriented Developers: An Issue of Skill Transfer." Journal of Object-Oriented Programming, Vol. 9, No. 7, pp. 6-10.
Merriam-Webster's Collegiate Dictionary. (2008), Merriam Webster Inc., Springfield, Mass.
Park, C., Crocker, C., Nussey, J., Springate, J., and Hutchings, D. (2010), "Evaluation of a Teaching Tool - Wiki - in Online Graduate Education." Journal of Information Systems Education, Vol. 21, No. 3, pp. 313-321.
Pennington, N., Lee, A., and Rehder, B. (1995), "Cognitive Activities and Levels of Abstraction in Procedural and Object-Oriented Design." Human-Computer Interaction, Vol. 10, No. 2, pp. 171-226.
Prajapati, B., Dunne, M., Bartlett, H., and Cubbidge, R. (2011), "The Influence of Learning Styles, Enrolment Status and Gender on Academic Performance of Optometry Undergraduates." Ophthalmic & Physiological Optics, Vol. 31, No. 1, pp. 69-78.
Ramesh, V. and Wu, J. (2004), "An Approach to Teaching Object Oriented Concepts in Business Schools." Decision Sciences Journal of Innovative Education, Vol. 2, No. 1, pp. 83-87.
Saljo, R. (2009), "Learning, Theories of Learning, and Units of Analysis in Research." Educational Psychologist; Vol. 44, No. 3, pp. 202-208.
Shroff, R., Vogel, D., and Coombes, J. (2008), "Assessing Individual-level Factors Supporting Student Intrinsic Motivation in Online Discussions: A Qualitative Study." Journal of Information Systems Education, Vol. 19, No. 1, pp. 111-126.
Sultan, F. and Chan, L. (2000), "The Adoption of New Technology: The Case of Object-Oriented Computing in Software Companies." IEEE Transactions on Engineering Management, Vol. 47, No. 1, pp. 106-126.
Syler, R., Cegielski, C., Oswald, S., and Rainer, R. (2006), "Examining Drivers of Course Perfromance: An Exploratory Examination of an Introductory CIS Applications Course." Decision Sciences Journal of Innovative Education, Vol. 4, No. 1, pp. 51-65.
Stefanidis, A. and Fitzgerald, G. (2010), "Mapping the Information Systems Curricula in UK Universities." Journal of Information Systems Education, Vol. 21, No. 4, pp. 391-409.
Strickland, D. S., Morrow, L. M., Girling-Butcher, W., Phillips, G., and Clay, M. (1991), "Fostering Independent Learning." The Reading Teacher, Vol. 44, No. 9, pp. 694-697.
Thorton, B., Haskell, H., and Libby, L. (2006), "A Comparison of Learning Styles between Gifted and Non-Gifted High School Students." Individual Differences Research, Vol. 4, No. 2, pp. 106-110.
Wang, Y. and Braman, J. (2009), "Extending the Classroom through Second Life." Journal of Information Systems Education, Vol. 20, No. 2, pp. 235-247.
Wang, Y. and Liao, H-C. (2011), "Adaptive Learning for ESL Based on Computation." British Journal of Educational Technology, Vol. 42, No. 1, pp. 66-87.
White, G. (2003), "Cognitive Characteristics of Learning C++." Journal of Computer Information Systems, Vol. 42, No. 3, pp. 51-55.
Yannakakis, G. N., Maragoudakis, M., and Hallam, J. (2009), "Preference Learning for Cognitive Modeling: A Case Study on Entertainment Preferences." IEEE

Transactions on Systems, Man & Cybernetics, Vol. 39, No. 6, pp. 1165-1175.
Yair, G. (2000), "Reforming Motivation: How the Structure of Instruction Affects Students' Learning Experiences." British Education Research Journal, Vol. 26, No. 2, pp. 191-210.
Zualkernan, I., Allert, J., and Qadah, G. (2006), "Learning Styles of Computer Programming Students: A Middle Eastern and American Comparison." IEEE Transactions on Education, Vol. 49, No. 4, pp. 443-450.
Zywno, M. S. (2003), "A contribution to validation of score meaning for Felder-Soloman's index of learning styles." Proceedings of the Society for Engineering Education Annual Conference and Exposition, Section 2351.

## AUTHOR BIOGRAPHIES

**Casey G. Cegielski**, Ph.D., CISA, CISSP, CRISC is an Associate Professor of Management Information Systems and former KPMG Faculty Fellow in the College of Business on the faculty of Auburn University in Auburn, Alabama. His current research interests are in the areas of innovation diffusion, emerging information technology, information security, and the strategic use of information technology. His research has appeared in several international information systems journals including Communications of the ACM, Information & Management, Decision Support Systems, and the Information Systems Journal. Additionally, Dr. Cegielski has more than 15 years of professional experience within the domain of information technology. He has served as a Senior Executive and an Executive Consultant in the financial, healthcare, and manufacturing sectors.



**Benjamin T. Hazen** is a PhD student in the Department of Management at Auburn University and an active duty US Air Force maintenance officer. His research interests include innovation and reverse logistics. His research has appeared in the proceedings of regional, national and international conferences and currently has several manuscripts under review at journals. The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defence, or the US Government.

**R. Kelly Rainer** is George Phillips Privett Professor of Management Information Systems at Auburn University, having received his Ph.D. from the University of Georgia. His research interests include enterprise resource planning (particularly using SAP), business process reengineering, and object-oriented technology. He recently spent a year with IBM’s Object Technology Practice. Prior to entering academia, Kelly was a practicing dentist for ten years.

Information Systems & Computing
Academic Professionals

Serving Information Systems Educators

## STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.



ISSN 1055-3096